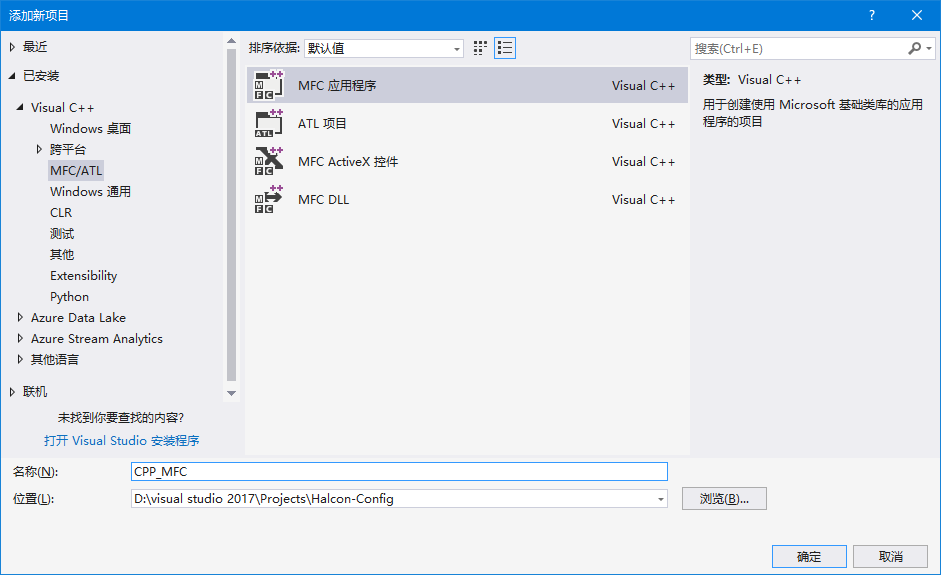
**Halcon与C++混合编程**

# 基础以及Halcon在MFC中的简单使用

1、新建一个MFC应用程序项目

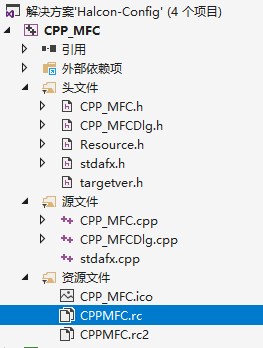


2、应用程序类型选择“基于对话框”，然后点击“完成”

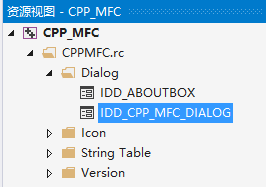
图片包含 屏幕截图

描述已自动生成

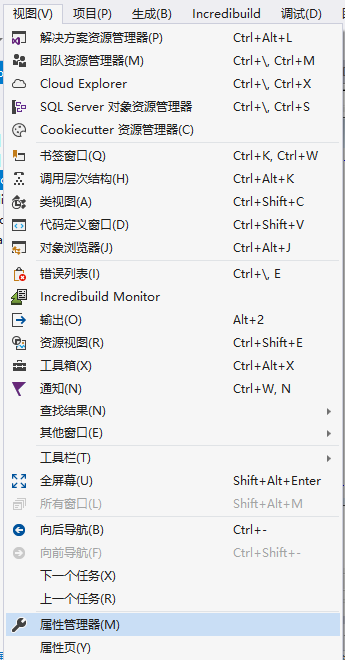
3、在新建的工程中双击CPPMFC.rc（名字可能有变化，主要看后缀）文件



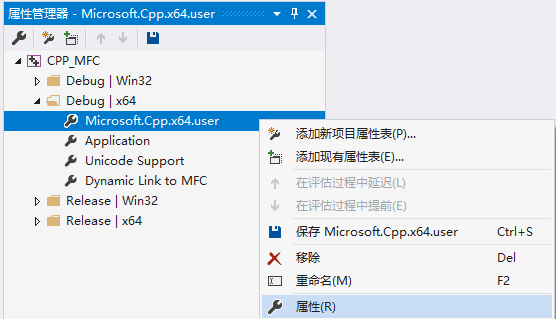
4、双击Dialog文件夹下的IDD\_CPP\_MFC\_DIALOG文件可以打开MFC的界面



5、点击“视图(V)”菜单中的“属性管理器(M)”，下面开始在C++中配置Halcon，本文中的方法配置后每次新建一个C++工程就不再需要重新配置了，网络上很多配置方法每次新建工程后需要重新配置



6、点击Debug|x64下的Microsoft.cpp.x64.user，右键点击“属性”，此处依据个人实际情况配置x86或者x64，只需要配置Debug类型就可以了，Release类型会自动配置好



7、选中“VC++目录”中的“包含目录”，然后点击“编辑”

图片包含 屏幕截图

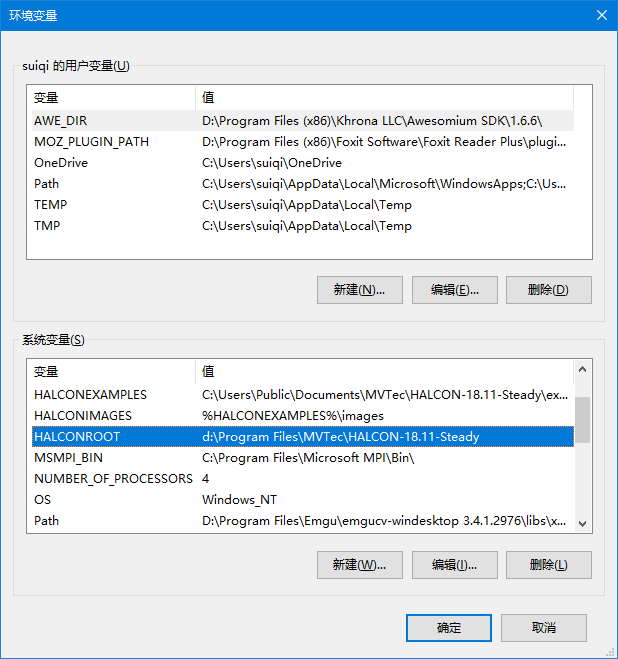
描述已自动生成

8、如下图中红色框或者绿色框标出的那样子，添加Halcon安装路径下的include文件夹以及include文件夹下的halconcpp文件夹

图片包含 屏幕截图

描述已自动生成

9、如上图红色框那样配置可以更具有通用性，因为安装Halcon时会自动在系统变量中添加其安装路径



10、选中“VC++目录”下的“库目录”，然后点击“编辑”

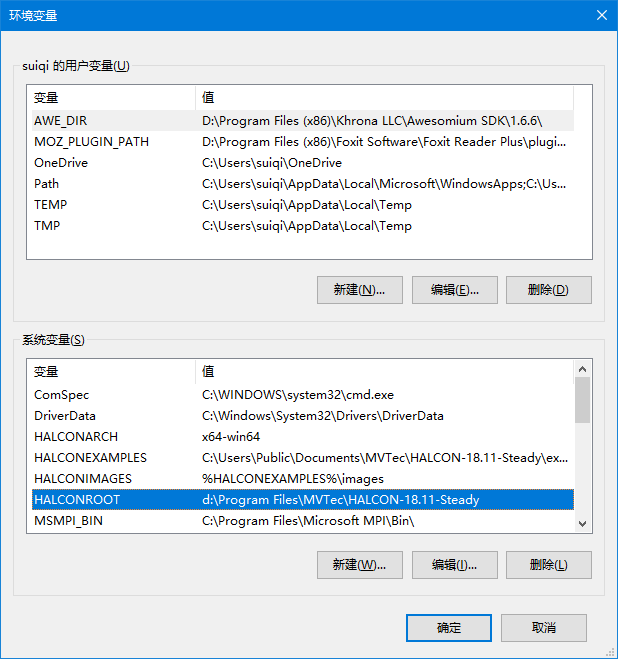
图片包含 屏幕截图

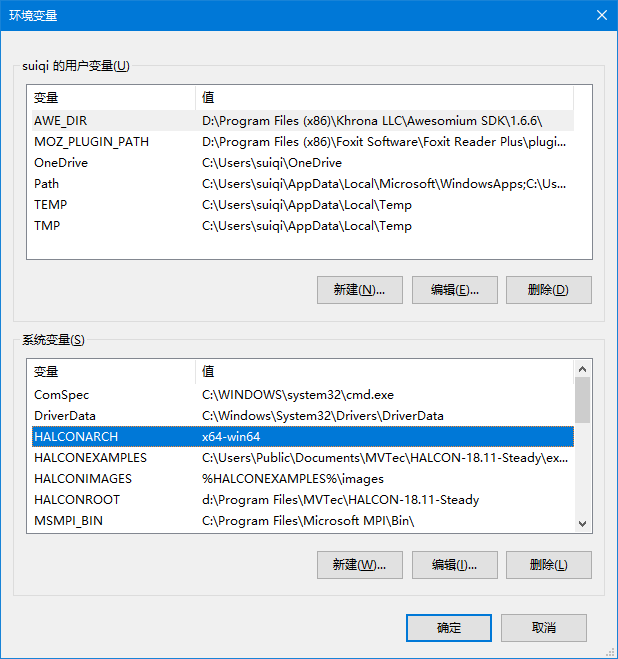
描述已自动生成

11、如下图红色框或者绿色框标出的那样，添加Halcon安装目录下的lib文件夹中的x64-win64，此处依据需要选择版本，如果前面选择配置的是Win32，那么此处做相应改变，红色框的配置同样具有通用性，这样子配置也同样是因为Halcon安装时自动在系统环境变量中添加了其路径及版本信息

图片包含 屏幕截图

描述已自动生成





12、点击“链接器”下的“输入”，选中“附加依赖项”，然后点击“编辑”，添加halconcpp.lib，然后点击“确定”，这样子在C++中使用Halcon就配置完成了

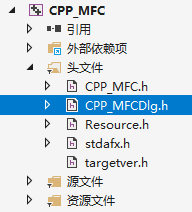
图片包含 屏幕截图

描述已自动生成

图片包含 屏幕截图

描述已自动生成

13、打开CPP\_MFC下的CPP\_MFCDlg.h文件（后面是Dlg.h，前面依据个人的工程名称变化）



14、在CPP\_MFCDlg.h文件（后面是Dlg.h，前面依据个人的工程名称变化）中添加代码

#include<HalconCpp.h>

#include<HALCONCpp/HDevThread.h>

using namespace HalconCpp;

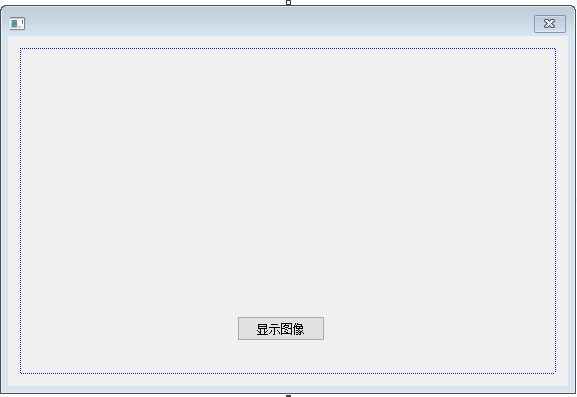
如下图所示

图片包含 屏幕截图

描述已自动生成

此处网络上有的网友写的是using namespace halcon;，可能之前的版本是这样子使用吧，最新的Halcon这样子书写是不正确的

15、在第4步中打开了MFC的界面，先将界面上的无关控件删掉，然后点击工具箱添加button控件，然后点击该控件，右键属性，在Caption中将其名字改成“显示图像”，**（按照此步骤继续后面显示的图像不在MFC窗口中，如果需要显示在MFC窗口中直接转到第22步）**如下图所示：

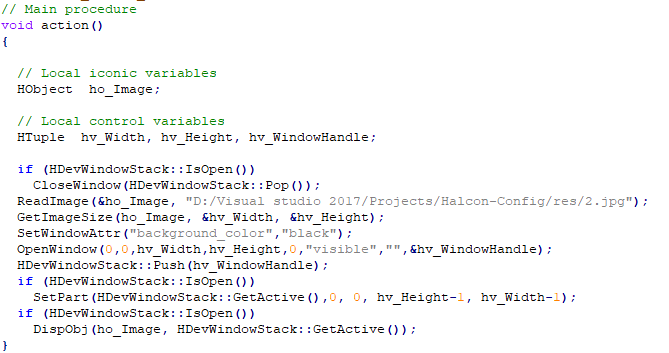


16、双击“显示图像”，在CPP\_MFCDlg.cpp(后面是Dlg.cpp，前面依据个人的工程名称变化）中Button控件的事件函数（本步骤中的双击动作会自动产生）中添加Halcon生成的CPP代码中的action函数中的内容，如下图所示：

图片包含 屏幕截图

描述已自动生成

下图所示就是Halcon生成的CPP源程序中的action函数

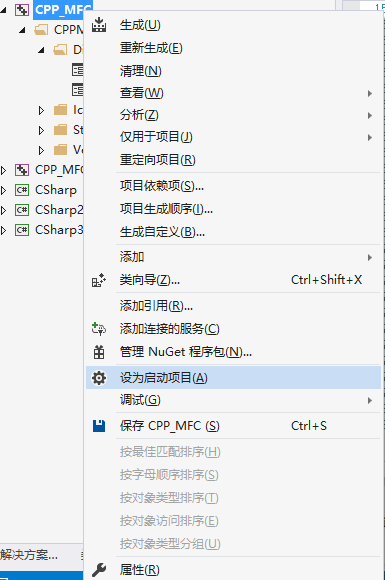


17、将上述添加的代码中的两句注释掉，否则会报错，报错原因暂时不清楚，如下图黑色框所示

图片包含 屏幕截图

描述已自动生成

18、右键单击项目名称，点击“设为启动项目(A)”

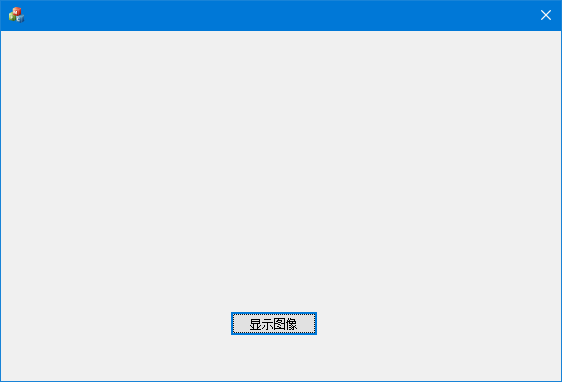


19、本文中操作之后结果如下图所示，点击“本地Windows调试器”

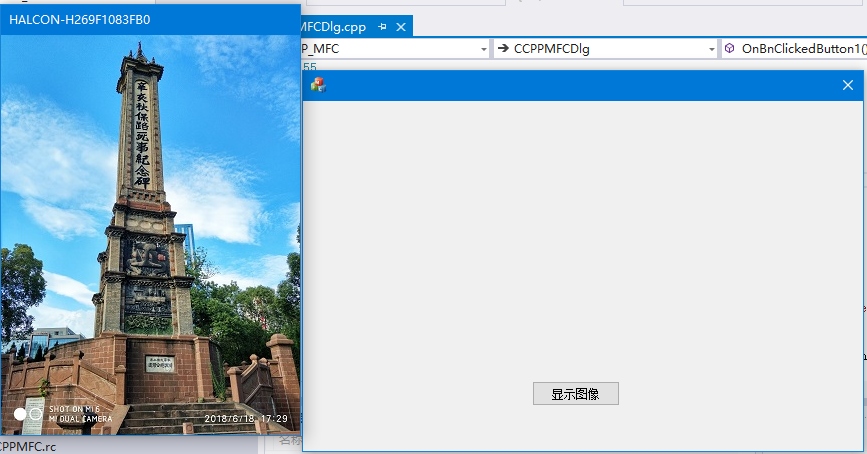
图片包含 屏幕截图

描述已自动生成

20、在下面的图中点击“显示图像”



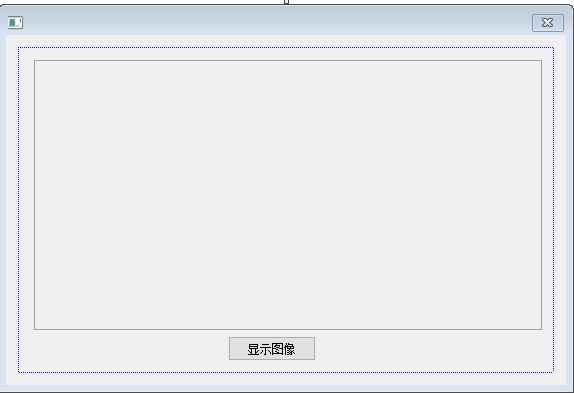
21、点击之后出现所要打开的图像，代表操作成功了



# Halcon及MFC使用的进阶

上述操作时图像显示在一个单独的窗口中，并不能显示在MFC窗口中，有必要进行额外的工作将其显示到MFC窗口中，如果需要显示到MFC窗口中，前面的步骤与基础部分前14步一致，下面是另外的操作

22、（接基础部分前14步）在第4步中打开了MFC的界面，先将界面上的无关控件删掉，然后点击工具箱添加Button控件，然后点击该控件，右键属性，在Caption中将其名字改成“显示图像”，然后添加Picture Control控件，如下图所示：

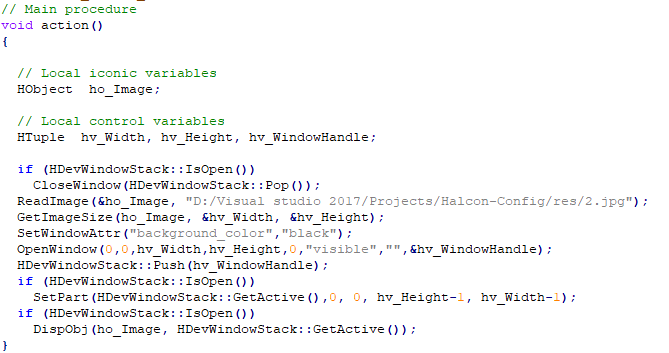


23、双击“显示图像”，在CPP\_MFCDlg.cpp(后面是Dlg.cpp，前面依据个人的工程名称变化）中Button控件的事件函数（本步骤中的双击动作会自动产生）中添加Halcon生成的CPP代码中的action函数中的内容，如下图所示：

图片包含 屏幕截图

描述已自动生成

下图所示就是Halcon生成的CPP源程序中的action函数



24、将下图中黑色框框出的语句注释掉，否则会报错，图中第三步

图片包含 屏幕截图

描述已自动生成

25、为了将图像显示到MFC窗体中的Picture Control控件中，需要得到其句柄，添加下图蓝色框框出的语句

//第四步，其中IDC\_STATIC为Picture Control的ID

HWND hImgWnd = GetDlgItem(IDC\_STATIC)->m\_hWnd;

如图中第四步

图片包含 屏幕截图

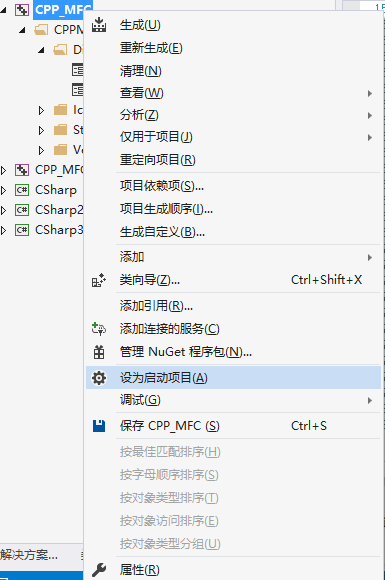
描述已自动生成

26、修改原先的打开窗口的语句，将其中的第三个0改成(LONG)hImgWnd，如图中黄色线和黄色框标注的地方

图片包含 屏幕截图

描述已自动生成

27、右键单击项目名称，点击“设为启动项目(A)”

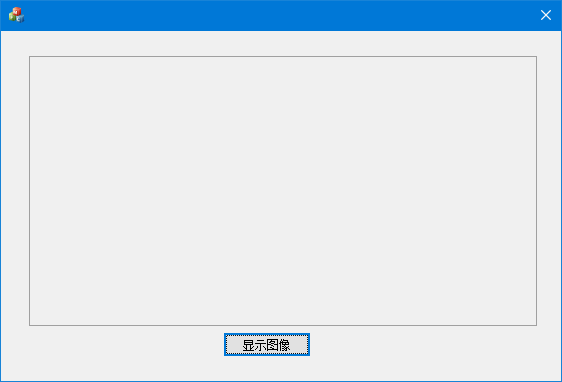


28、本文中操作之后结果如下图所示，点击“本地Windows调试器”

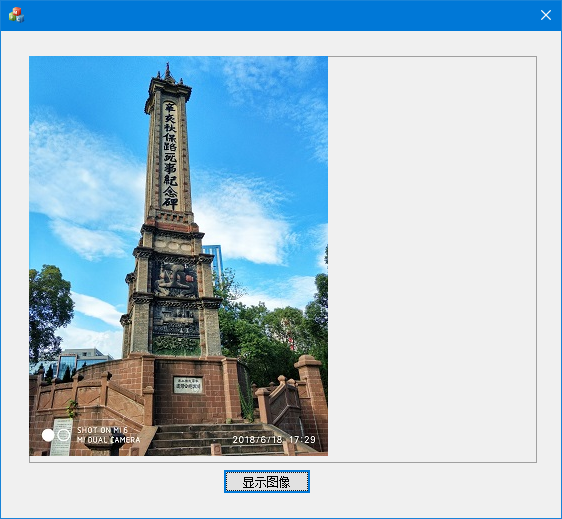
图片包含 屏幕截图

描述已自动生成

29、在下面的图中点击“显示图像”



30、点击之后出现所要打开的图像，代表操作成功了



图像中的Picture Control暂时未找到怎么设置可以自适应图像大小添加滚动条

# Halcon代码在C++控制台程序中的应用

有时候不需要利用MFC窗口，只需要一个控制台程序即可，操作步骤如下

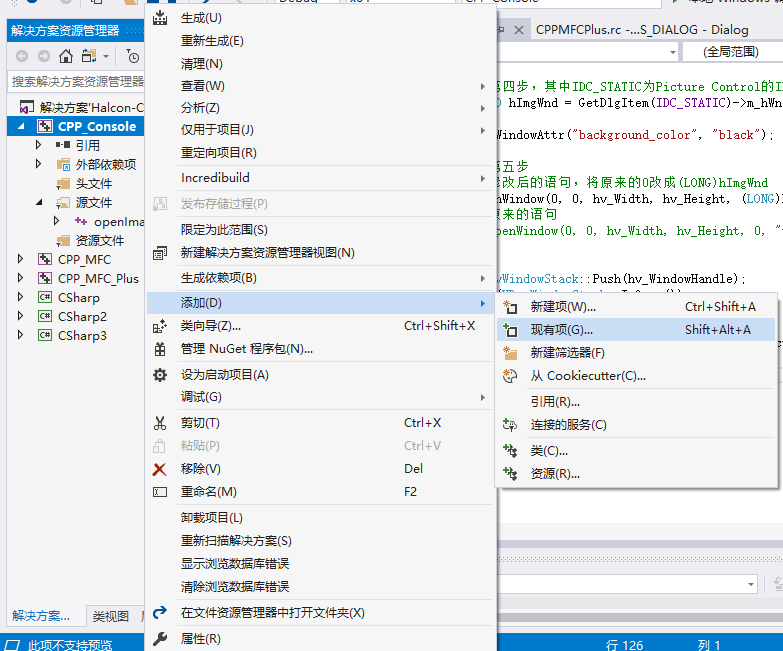
1）、如下图所示，新建一个Visual C++空项目

图片包含 屏幕截图

描述已自动生成

2）、Halcon及C++的配置参考前面的第5到第12步

3）、右键单击工程名，然后点击“添加(D)”中的“现有项(G)”，选中Halcon导出的C++源程序



4）、如下图所示，选中源程序，点击“添加(A)”

图片包含 屏幕截图

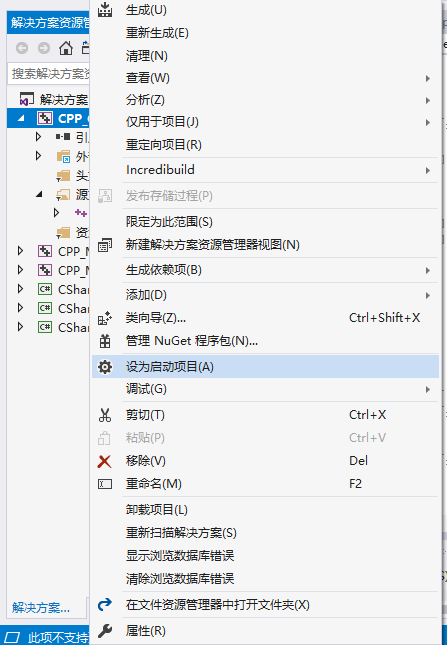
描述已自动生成

5)、找到导入的源程序，如下图所示在main函数最后面的语句return ret;前面添加代码getchar();或者system("pause");，否则运行程序时会一闪而过

图片包含 屏幕截图

描述已自动生成

6）、右键单击项目名称，点击“设为启动项目(A)”



7）、本文中操作之后结果如下图所示，点击“本地Windows调试器”

图片包含 屏幕截图

描述已自动生成

8）、运行结果如下图所示



9）、在黑色的控制台上面输入任意键即可退出程序